Solr.xml 4.4 and beyond

Solr.xml 4.4 and beyond

As of ⚠️ Solr4.4 there is an alternate structure for the solr.xml config file **which will become mandatory for 5.0**.

- Optionally as of ⚠️ Solr4.4 and mandatory for Solr5.0, the structure of the solr.xml file has changed. In a nutshell, <cores> and <core> have been replaced by auto-discovering cores. Whether to use old or new-style core definitions is determined by whether the <cores> tag is present in solr.xml. In 5.0, presence of the <cores> tag will generate an error on startup.
- An optional property *coreRootDirectory* can cause the discovery process to start at an arbitrary directory other than SOLR_HOME.
- We'll distribute a new-style solr.xml as the default in the example directory with Solr4.4
- The *sharedLib* attribute on the top-level <*solr*> element is replaced by a child *str* element (see below). In theory, the old mechanism should continue to support *sharedLib*, but SOLR-4791 documents the fact that it doesn't work in 4.3.0.

As of ⚠️ Solr4.5, solr.xml may be stored on your ZooKeeper ensemble, see SOLR-4718.

Structure of the new-style solr.xml

Basically, it's mostly a move from the older attributes to a flatter style, and removal of <cores> and <core> tags. Here's a sample file. Note that any of these values can have a system property defined by specifying -Dpropname=propvalue on JVM startup:

Any of these will honor system property substitution following the usual rules of ${propname:default_value}. Remember that a form ${sysprop:} will use the built-in defaults and can be omitted from the config, they're included here to document that they're available. Where it's straightforward to find in the code, the defaults are included. Note that the defaults are subject to change.

⚠️ This example has many options listed here for reference. You should not change them unless and until you have a need. Start with the solr.xml in the example directory in the distro (<solrHome>/example/solr/solr.xml) and only add in specific options as the need arises.

```
<solr>
  <str name="adminHandler">${adminHandler:org.apache.solr.handler.admin.CoreAdminHandler}</str>
  <int name="coreLoadThreads">${coreLoadThreads:3}</int>
  <str name="coreRootDirectory">${coreRootDirectory:}</str> <!-- usually solrHome  -->
  <str name="managementPath">${managementPath:}</str>
  <str name="sharedLib">${sharedLib:}</str>
  <str name="shareSchema">${shareSchema:false}</str>
  <int name="transientCacheSize">${transientCacheSize:Integer.MAX_VALUE}</int> <!-- ignored unless cores are
defined with transient=true -->

  <solrcloud>
    <int name="distribUpdateConnTimeout">${distribUpdTimeout:}</int>
    <int name="distribUpdateSoTimeout">${distribUpdateTimeout:}</int>
    <int name="leaderVoteWait">${leaderVoteWait:}</int>
    <str name="host">${host:}</str>
    <str name="hostContext">${hostContext:solr}</str>
    <int name="hostPort">${jetty.port:8983}</int>
    <int name="zkClientTimeout">${zkClientTimeout:15000}</int>
    <str name="zkHost">${zkHost:}</str>
    <bool name="genericCoreNodeNames">${genericCoreNodeNames:true}</bool>
  </solrcloud>


  <logging>
    <str name="class">${loggingClass:}</str>
    <str name="enabled">${loggingEnabled:}</str>
    <watcher>
      <int name="size">${loggingSize:}</int>
      <int name="threshold">${loggingThreshold:}</int>
    </watcher>
  </logging>

  <shardHandlerFactory name="shardHandlerFactory" class="HttpShardHandlerFactory">
    <int name="socketTimeout">${socketTimeout:}</int>
    <int name="connTimeout">${socketTimeout:}</int>
  </shardHandlerFactory>

</solr>
```

Core discovery process

See: Core Discovery (4.4 and beyond).

Core discovery happens at startup. Exploration of the core tree terminates when a file named core.properties is encountered. Discovery of a file of that name is assumed to define the root of a core. There is no a-priori limit on the depth of the tree. That is, the directories under the core root are explored until a core.properties file is encountered, and that directory is assumed to be the instanceDir for that core. Subdirectories of any directory that has a core.properties file are NOT examined for additional cores. The core.properties file presently recognizes the following entries:

- name - the name of the core. If not specified, the name comes from the containing directory.
- config - the configuration file. Defaults to solrconfig.xml
- dataDir - the directory where the index, tlog, etc. are stored. Again, since this is discovery-based, omit this unless you have special needs.
- ulogDir - where the transaction log resides. It may be advantageous to put the transaction lot on a different disk than the index.
- schema - the schema file. Defaults to schema.xml
- shard - the shard ID.
- collection - the collection to which this core belongs
- roles - SolrCloud role definition
- properties - properties file to override core definitions. TBD: This is probably obsolete since we're reading a properties file in the first place. Is there a use case for supporting this now?

- loadOnStartup - [**true**|false] this core should be loaded and a searcher opened when Solr starts.

- transient - [true|**false**] this core may be unloaded if the core cache exceeds transientCacheSize (defined in solr.propreties)

- coreNodeName - SolrCloud core node name

Minimal core.properties file

Interestingly, the minimal core.properties file is empty. Say an empty core.properties file is discovered in /solr/home/core1. The name will default to "core1", the instanceDir to /solr/home/core1, the dataDir to /solr/home/core1/data etc.

Relation with CoreAdmin api

The core discovery process happens only at startup. After startup, Solr does not monitor changes in the core root directory.

- to add a new core, create an instanceDir in the core root directory for the core without a core.properties file, and use the CREATE CoreAdmin operation
- to remove an existing core, use the UNLOAD CoreAdmin operation

Ignoring cores

It's not hard to imagine that people will want to have the core discovery process temporarily ignore one or more cores. The easiest way to accomplish this would be to rename the core.properties file, e.g. core.properties.bak.