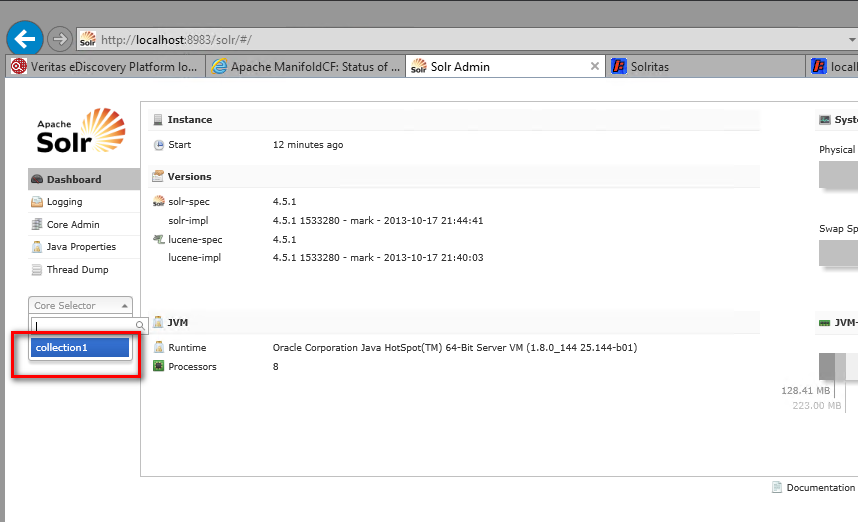
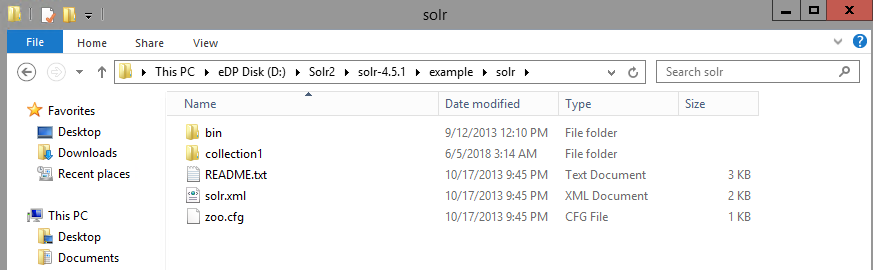
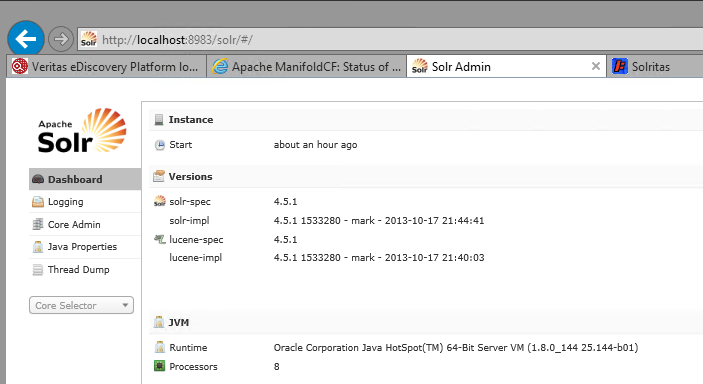
**How to add an additional Solr core (i.e. an additional index)**  
**Note:** These steps are no longer required for newer Solr versions. Newer Solr versions use “solr.cmd create\_core”. See my DocSearcher and \_solr\_lib.bat.  
  
From SolrInAction:  
*“..each core as a separate index and configuration, and there can be many cores in a single Solr instance. One use of Solr’s multicore support is data partitioning, such as having one core for recent documents and another core for older documents, known as chronological sharding. Another use of Solr’s multicore support is to support multitenant applications.”  
  
“In Solr, a core is composed of a set of configuration files, Lucene index files, and Solr’s transaction log. One Solr server running in Jetty can host multiple cores.”  
  
“As of Solr 4.4, cores can be autodiscovered and do not need to be defined in solr.xml”.  
(Solr also provides a Core Admin API that allows you to create, update, and delete cores programmatically from your application. We cover the Core Admin API in more detail in chapter 12.)*  
The default Solr “example” instance – A single core named “collection1”:  
  
  
This core named “collection1” exists here as a directory in the “example/solr/” folder:  
  
  
(**Note:** To confuse things a little, Solr also uses the term “collection” in the context of a Solr cluster, in which a single index is distributed across multiple servers.)

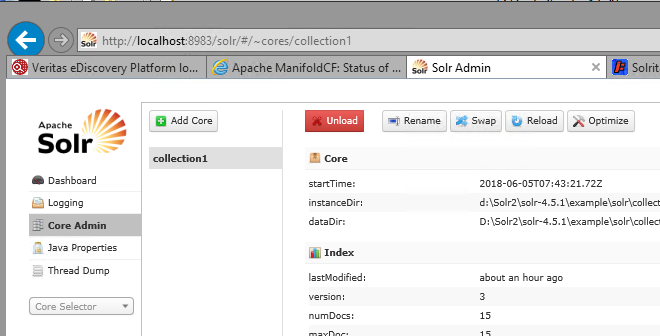
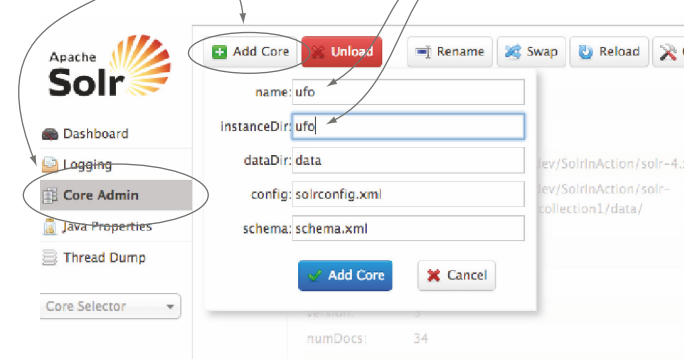
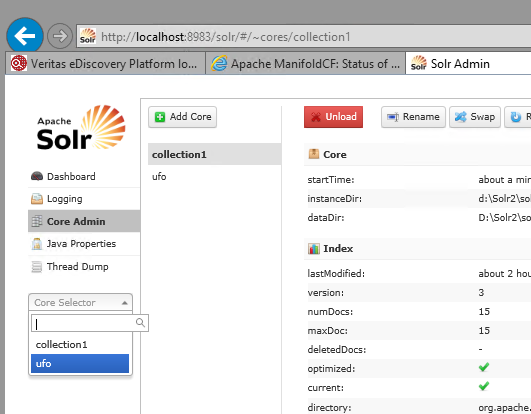
**Solr home directory**Each Solr server has one and only one Solr home directory that contains all cores. The global Java system property **solr.solr.home** sets the location of the Solr home directory.   
  
Because we use “java –jar start.jar” to start Solr, we won’t see “solr.solr.home” defined anywhere in a file. From “example/README.txt”:  
 *“By default, start.jar starts Solr in Jetty using the default Solr Homedirectory of "./solr/" (relative to the working directory of hte servlet container).”*  
This causes the property to be automatically set to “$SOLR\_INSTALL/example/solr/”.  
  
  
**core.properties***“Solr scans the Solr home directory for subdirectories containing a core.properties file, which defines basic properties for autodiscovered cores. For instance, the example server has a core.properties file in the example/solr/collection1/ directory. The core.properties file contains a single line defining the name of the core, name=collection1, which is all that is needed to trigger autodiscovery for the collection1 core.”*

**Adding a Core from the Admin page**(See page 284 of SolrInAction: “Set up a new Solr core for UFO sightings”).1) **Stop Solr** and then copy the “collection1/” directory under “$SOLR\_INSTALL/example/” to a new directory named “ufo/”:

|  |
| --- |
| cd $SOLR\_INSTALL/example/solr  cp -r collection1 ufo  rm -rf ufo/data *­­* rm ufo/core.properties |

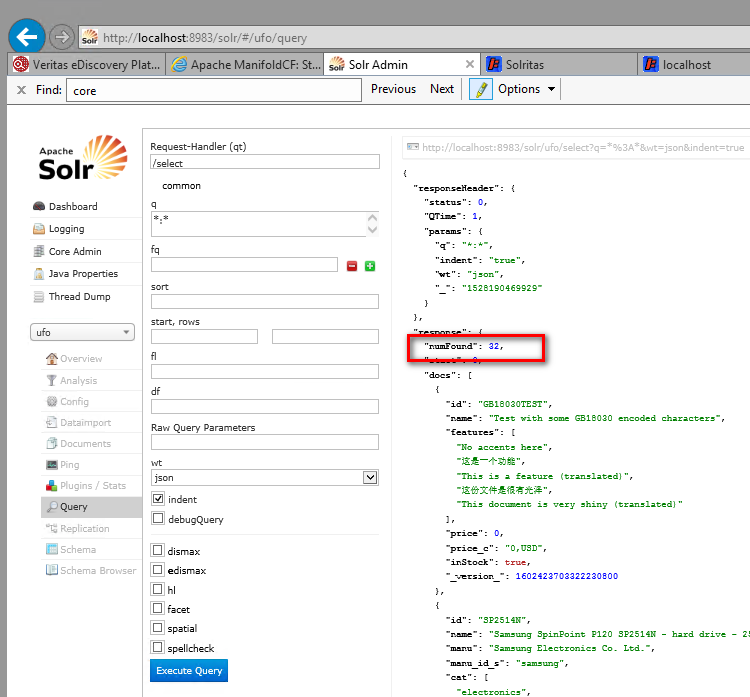
2) **Start Solr** (java –jar start.jar) and then navigate to <http://localhost:8983/solr> and click “Core Admin” in the left-hand panel:  


3) Click the “Add Core” button:

4) Fill out the form like this:  
This will create a core.properties file for a new core.  
  
  
Our new Core is now listed in the Solr Admin UI:  


**post.jar –directing document updates to our new Core**FYI: It looks like pre-v5, Solr had a notion of a “default core”, and the post.jar (SimplePostTool.java) in Solrv4 is taking advantage of this.  
  
But here’s how we can still direct documents to our new “ufo” Core, using the Solrv4 post.jar tool:

|  |
| --- |
| $ cd example/exampledocs $ **java -Durl="http://localhost:8983/solr/ufo/update" -jar post.jar \*.xml**  SimplePostTool version 1.5  Posting files to base url http://localhost:8983/solr/ufo/update using content-type application/xml..  POSTing file gb18030-example.xml  POSTing file hd.xml  POSTing file ipod\_other.xml  POSTing file ipod\_video.xml  POSTing file manufacturers.xml  POSTing file mem.xml  POSTing file money.xml  POSTing file monitor.xml  POSTing file monitor2.xml  POSTing file mp500.xml  POSTing file sd500.xml  POSTing file solr.xml  POSTing file utf8-example.xml  POSTing file vidcard.xml  14 files indexed.  COMMITting Solr index changes to http://localhost:8983/solr/ufo/update..  Time spent: 0:00:00.327 |

This results in 32 documents into our Core:  
$ grep '<doc>' \*.xml | wc –l  
32  
  


*JeremyC 5-6-2018***END**