

SolrTomcat

{{{#!wiki red/solid Beginning with Solr 5.0, Solr is no longer distributed as a "war" (Web Application Archive) suitable for deployment in any Servlet Container. Solr is now distributed as a stand alone java server application, including start and stop scripts for use on Unix and MS-Windows platforms, as well as an installation script for setting up a "production" installation of Solr on *nix platforms managed via /etc/init.d. See <https://wiki.apache.org/solr/WhyNoWar> }}}

Solr with Apache Tomcat

Solr has been tested on [Tomcat](#) 5.5, 6, and 7. In Tomcat 7 there was a bug with resolving URLs ending in "/". This should be fixed in Tomcat 7.0.5+, see [S OLR-2022](#) for full details.

See the instructions in the generic [Solr installation](#) page for general info before consulting this page.

- [Solr with Apache Tomcat](#)
 - [Simple Example Install](#)
 - [Installing Tomcat 6](#)
 - [Building Solr](#)
 - [Installing Solr instances under Tomcat](#)
 - [Single Solr Instance](#)
 - [Troubleshooting](#)
 - [Login to Tomcat Management page does not work](#)
 - [Tomcat Manager does not list Solr](#)
 - [Exceptions when visiting Solr admin](#)
 - [HTTP 500 error](#)
 - [Optional Configuration](#)
 - [Logging](#)
 - [URI Charset Config \(only needed before Solr 4.1\)](#)
 - [Configuring Solr Home with JNDI](#)
 - [Enabling Longer Query Requests](#)
 - [Multiple Solr Webapps](#)
 - [Tomcat on Windows](#)
 - [Single Solr app](#)
 - [Multiple Solr apps](#)
 - [64-bit Note](#)
 - [//!\ TODO //!](#)
 - [External Resources](#)
 - [Troubleshooting Errors](#)
 - [Solr Tomcat Distributions](#)
 - [HDS - Heliosearch Distribution for Solr](#)

Simple Example Install

⚠ [Solr4.3](#) requires completely different deployment. These instructions are **not** current and are for an indeterminate version of Solr.

⚠ Though this page needs to be completely re-written for the latest Solr version, here are the main differences with Solr 4.3 (at least for running a single instance). These steps are required for both Linux and Windows.

- Java 1.7 is required
- The JAR files from the Solr lib/ext directory (something like /opt/solr/example/lib/ext) must be copied to \$CATALINA_HOME/lib/
- The log4j.properties file from the resources file (something like /opt/solr/example/resources) must be copied to \$CATALINA_HOME/lib/

Installing Tomcat 6

[Apache Tomcat](#) is a web application server for Java servlets. These are instructions for manually installing Tomcat 6 on Linux, recommended because distribution Tomcats are either old or quirky.

Create the *solr* user. As *solr*, extract the [Tomcat 6.0 download](#) into /opt/tomcat6, hereafter referred to as the \$CATALINA_HOME directory.

Edit \$CATALINA_HOME/conf/tomcat-users.xml to enable the manager login as user "tomcat" with password "tomcat" (insecure):

```
<role rolename="manager"/>
<role rolename="admin"/>
<user username="tomcat" password="tomcat" roles="manager,admin"/>
```

Start Tomcat with \$CATALINA_HOME/bin/catalina.sh run. Tomcat runs on the port defined \$CATALINA_HOME/conf/server.xml, configured by default to port 8080.

The startup script [tomcat6](#) can be placed in `/etc/init.d/tomcat6` on CentOS/RedHat/Fedora so that you can start Tomcat using `service tomcat6 start`. Use `chkconfig` to enable the tomcat6 service to start on boot.

Building Solr

Skip this section if you have a binary distribution of Solr. These instructions are for building [Solr](#) from source, if you have a nightly tarball or have checked out the trunk from subversion at <http://svn.apache.org/repos/asf/lucene/dev/trunk>. Assumes that you have [JDK 1.6](#) already installed.

In the source directory, run `ant dist` to build the `.war` file under `dist`. Build the example for the [Solr tutorial](#) by running `ant example`. Change to the 'example' directory, run `java -jar start.jar` and visit `localhost:8983/solr/admin` to test that the example works with the Jetty container.

Installing Solr instances under Tomcat

Assuming that Solr and its `example` are built, this is how to install the Solr example as an instance under Tomcat.

Copy the `example/solr` directory from the source to the installation directory like `/opt/solr/example/solr`, hereafter `$SOLR_HOME`. Copy the `.war` file `dist/apache-solr-*.war` into `$SOLR_HOME` as `solr.war`.

The configuration file `$SOLR_HOME/conf/solrconfig.xml` in the example sets `dataDir` for the index to be `./solr/data` relative to the current directory - which is true for running the Jetty server provided with the example, but incorrect for Tomcat running as a service. Modify the `dataDir` to specify the full path to `$SOLR_HOME/data`:

- ```
<dataDir>${solr.data.dir:/opt/solr/example/solr/data}</dataDir>
```

The `dataDir` can also be temporarily overridden with the `JAVA_OPTS` environment variable prior to starting Tomcat:

- ```
export JAVA_OPTS="$JAVA_OPTS -Dsolr.data.dir=/opt/solr/example/solr/data"
```

Create a Tomcat Context fragment to point `docBase` to the `$SOLR_HOME/solr.war` file and `solr/home` to `$SOLR_HOME`:

```
<?xml version="1.0" encoding="utf-8"?>
<Context docBase="/opt/solr/example/solr/solr.war" debug="0" crossContext="true">
  <Environment name="solr/home" type="java.lang.String" value="/opt/solr/example/solr" override="true"/>
</Context>
```

Symlink or place the file in `$CATALINA_HOME/conf/Catalina/localhost/solr-example.xml`, where Tomcat will automatically pick it up. Tomcat deletes the file on undeploy (which happens automatically if the configuration is invalid).

Repeat the above steps with different installation directories to run multiple instances of Solr side-by-side.

If Tomcat is not already running, start it with `service tomcat6 start` or `$CATALINA_HOME/bin/startup.sh run`. The Solr admin should be available at `http://<host>:8080/solr-example/admin`.

Single Solr Instance

If you are sure that you will only ever run one instance of Solr, you can do away with the Context fragment by placing the `.war` in `$CATALINA_HOME/webapps/solr-example.war` and setting the Solr home through a global environment variable prior to starting Tomcat:

```
export JAVA_OPTS="$JAVA_OPTS -Dsolr.solr.home=/opt/solr/example"
```

Troubleshooting

Login to Tomcat Management page does not work

`$CATALINA_HOME/conf/tomcat-users.xml` may be missing the correct user line.

Tomcat Manager does not list Solr

The Context fragment may be invalid. Examine `$CATALINA_HOME/tomcat6/logs/catalina.out`.

Exceptions when visiting Solr admin

View `$CATALINA_HOME/logs/catalina.out` for a better view of the exceptions. Probably caused by an incorrect path in `solrconfig.xml` or the Context fragment, or by an unclean build (run *ant clean* and rebuild the source).

HTTP 500 error

If, when installing Solr 3.5, you get an HTTP 500 error and the exception message begins with

```
org.apache.solr.common.SolrException: Error loading class 'solr.VelocityResponseWriter' at org.apache.solr.core.SolrResourceLoader.findClass(SolrResourceLoader.java:389)...
```


the problem is caused by incorrect `<lib>` locations in `$SOLR_HOME/conf/solrconfig.xml` and can be fixed by reading [The Solr 3.5 Post-Release Notes](#).

Optional Configuration

Logging

For information about controlling JDK Logging (aka: java.util logging) in Tomcat, please consult the Tomcat docs... <http://tomcat.apache.org/tomcat-6.0-doc/logging.html>

URI Charset Config (only needed before Solr 4.1)

 **Solr4.1** Solr now parses request parameters (in URL or sent with POST using content-type `application/x-www-form-urlencoded`) in its dispatcher code. It no longer relies on special configuration settings in Tomcat or other web containers to enable UTF-8 encoding, which is mandatory for correct Solr behaviour.

Solr now works out of the box with e.g. Tomcat, JBoss,...

In older Solr versions (before [Solr4.1](#)), you have to revisit this: Edit Tomcat's `conf/server.xml` and add the following attribute to the correct Connector element: `URIEncoding="UTF-8"`.

```
<Server ...>
  <Service ...>
    <Connector ... URIEncoding="UTF-8" />
    ...
  </Connector>
</Service>
</Server>
```

Configuring Solr Home with JNDI

A Tomcat context fragments can be used to configure the JNDI property needed to specify your Solr Home directory.

Just put a context fragment file under `$CATALINA_HOME/conf/Catalina/localhost` that looks something like this...

```
$ cat /tomcat55/conf/Catalina/localhost/solr.xml
```

```
<Context docBase="/some/path/solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="/my/solr/home" override="true" />
</Context>
```

A few things to keep in mind:

- The `"conf/Catalina/localhost"` directory may not exist by default in your installation. You may have to create it.
- `"/some/path/solr.war"` is the absolute path to where ever you want to keep the Solr war using the appropriate syntax for your Operating System. In Tomcat 5.5 and later, the war file must be stored outside of the webapps directory for this to work. Otherwise, this entire Context element is ignored.
- `"/my/solr/home"` should be to where you have created your Solr Home directory, using the appropriate syntax for your Operating System.
- Prior to Tomcat 5.5, a `"path"` attribute was required for Context elements (starting with 5.5, the path attribute must not be used except when statically defining a Context in `server.xml`, as it will be inferred from the Context fragment filename).

Enabling Longer Query Requests

If you try to submit too long a GET query to Solr, then Tomcat will reject your HTTP request on the grounds that the HTTP header is too large; symptoms may include an HTTP 400 Bad Request error or (if you execute the query in a web browser) a blank browser window.

If you need to enable longer queries, you can set the `maxHttpHeaderSize` attribute on the HTTP Connector element in your `server.xml` file. The default value is 4K. (See <http://tomcat.apache.org/tomcat-5.5-doc/config/http.html>)

Multiple Solr Webapps

Tomcat context fragments make configuring multiple Solr webapps (with JNDI) in a single Tomcat server easy.

Just follow the previous instructions for "Configuring Solr Home with JNDI" to create a separate context fragment file under \$CATALINA_HOME/conf/Catalina/localhost for each solr webapp you want to run:

```
$ cat /tomcat55/conf/Catalina/localhost/solr1.xml
```

```
<Context docBase="/some/path/solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="/some/path/solr1home" override="true" />
</Context>
```

```
$ cat /tomcat55/conf/Catalina/localhost/solr2.xml
```

```
<Context docBase="f:/solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="/some/path/solr2home" override="true" />
</Context>
```

Don't put anything related to Solr under the webapps directory.

The solr home directories are configured via JNDI in the context fragment, and in the examples above will be /some/path/solr1home and /some/path/solr2home. The URLs to the two webapps will be <http://host:port/solr1> and <http://host:port/solr2>

Tomcat on Windows

Single Solr app

- Download and install [Tomcat](#) for Windows using the MSI installer. Install it with the tcnative.dll file. Say you installed it in c:\tomcat\
- Check if Tomcat is installed correctly by going to <http://localhost:8080/>
- Change the c:\tomcat\conf\server.xml file to add the URLEncoding Connector element as shown above.
- Download and unzip the Solr distribution zip file into (say) c:\temp\solrZip\
- Make the "solr home" directory called, where you intend the application server to function, say c:\web\solr\
- Copy the contents of the example\solr directory c:\temp\solrZip\example\solr\ to c:\web\solr\
- Stop the Tomcat service
- Copy the *solr*.war file from c:\temp\solrZip\dist\ to the Tomcat webapps directory c:\tomcat\webapps\
- Rename the *solr*.war file solr.war
- Configure Tomcat to recognize the solr home directory you created, by adding the Java Options -Dsolr.solr.home=c:\web\solr and -Dsolr.velocity.enabled=false
 - either use the system tray icon to add the java option
 - or manually edit the environment script c:\tomcat\bin\setenv.bat and add it to JAVA_OPTS
- * Note: For Tomcat 7 and Solr 3.4 (last version on 2011-09-23), the above option on setenv.bat may not work, so you could not use it and put this code fragment on \$CATALINA_HOME/conf/Catalina/localhost/solr.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Context docBase="C:\apache-tomcat-7.0.21\webapps\solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="C:\solr" override="true" />
</Context>
```

- Start the Tomcat service
- Go to the solr admin page to verify that the installation is working. It will be at <http://localhost:8080/solr/admin>
- NOTES: The following are notes from a novice installer on Windows 7, Tomcat 7.0.50, and Solr 4.7.0. I had help from <http://stackoverflow.com/questions/17619809/installing-apache-solr-4-3-1-on-apache-tomcat-6-0>. I'm not sure everything is elegant, but this is how I got it working.
 - Follow the instructions above.
 - Logging Jars: Make sure to copy the logging jars from C:\solr-4.7.0\example\lib\ext to the Tomcat lib directory (such as C:\apache-tomcat-7.0.50\lib) or you will likely get a "severe error filterstart" error when starting Tomcat.
 - Folder Structure: In my case, I named the root folder C:\solr-4.7.0, I made a subfolder called C:\solr-4.7.0\solr_home, and I copied the contents of C:\solr-4.7.0\example\solr directly into C:\solr-4.7.0\solr_home. So inside C:\solr-4.7.0\solr_home there should be "bin" and a "collection1" subfolders as well as several files.
 - Tomcat Startup: I started tomcat using C:\apache-tomcat-7.0.50\bin\startup.bat. I modified this file by adding the line "set JAVA_OPTS=-Dsolr.solr.home=c:\solr-4.7.0\solr_home -Dsolr.velocity.enabled=false" and things worked.
 - So far I have not needed to modify the URLEncoding Connector as mentioned above. I believe that may not be necessary with solr-4.7.0+, but not sure.

- Tomcat by default runs on port 8080. Solr includes a Java application called post.jar for posting data to the index, which is referenced in the tutorial. It can be found in C:\solr-4.7.0\example\exampledocs. Make sure that you pass this application the correct port number as by default it expects port 8983 and you'll get a connection error. For example, at a windows command prompt, "java -Durl=<http://localhost:8080/solr/update> -jar post.jar hd.xml"

Multiple Solr apps

- Download and install [Tomcat](#) for Windows using the MSI installer. Install it with the tcnative.dll file. Say you installed it in c:\tomcat\
- Check if Tomcat is installed correctly by going to <http://localhost:8080/>
- Change the c:\tomcat\conf\server.xml file to add the URLEncoder Connector element as shown above.
- Download and unzip the Solr distribution zip file into (say) c:\temp\solrZip\
- Say you need two apps in c:\web\solr1 and c:\web\solr2; create these two directories
- Copy the contents of the example\solr directory c:\temp\solrZip\example\solr\ to c:\web\solr1\ and to c:\web\solr2\
- Stop the Tomcat service
- Copy the *solr*.war file from c:\temp\solrZip\dist\ to the Tomcat lib directory c:\tomcat\lib\
- Rename the *solr*.war file solr.war
- Make a new text file in c:\tomcat\conf\Catalina\localhost called solr1.xml with the following code fragment

```
<Context docBase="c:\tomcat\lib\solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="c:\web\solr1" override="true" />
</Context>
```

- Make a new text file in c:\tomcat\conf\Catalina\localhost called solr2.xml with the following code fragment

```
<Context docBase="c:\tomcat\lib\solr.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String" value="c:\web\solr2" override="true" />
</Context>
```

- Start the Tomcat service
- Go to the solr admin pages for the 2 webapps to verify that the installation is working. It will be at <http://localhost:8080/solr1/admin> and <http://localhost:8080/solr2/admin>

64-bit Note

The MSI installer that installs Tomcat as a Windows service isn't prepared to support 64-bit Windows out of the box. There are some straightforward workarounds, though. See <http://stackoverflow.com/questions/211446/how-to-run-tomcat-6-on-winxp-64-bit>

!!!\ TODO ///

Indicate how to index in tomcat (rather than built-in jetty support via start.jar).

To use the indexing examples in the Solr Tutorial, use the -Durl parameter, e.g.

```
java -jar -Durl=http://localhost:8080/solr/update post.jar solr.xml monitor.xml
```

External Resources

<http://www.ibm.com/developerworks/java/library/j-solr1/>

Troubleshooting Errors

It's possible that you get an error related to the following:

```
SEVERE: Exception starting filter SolrRequestFilter
java.lang.NoClassDefFoundError: Could not initialize class org.apache.solr.core.SolrConfig
    at org.apache.solr.servlet.SolrDispatchFilter.init(SolrDispatchFilter.java:76)
.....
Caused by: java.lang.RuntimeException: XPathFactory#newInstance() failed to create an XPathFactory for the
default object model: http://java.sun.com/jaxp/xpath/dom with the XPathFactoryConfigurationException: javax.xml.
x
path.XPathFactoryConfigurationException: No XPathFactory implementation found for the object model: http://java.
sun.com/jaxp/xpath/dom
    at javax.xml.xpath.XPathFactory.newInstance(Unknown Source)
```

This is due to your tomcat instance not having the xalan jar file in the classpath. It took me some digging to find this, and thought it might be useful for others. The location varies from distribution to distribution, but I essentially just added (via a symlink) the jar file to the shared/lib directory under the tomcat directory.

Solr Tomcat Distributions

HDS - Heliosearch Distribution for Solr

Heliosearch Distribution for Solr contains a pre-configured Tomcat based Solr server that is open source and free to use.

<http://heliosearch.com/heliosearch-distribution-for-solr/>

Simply download and go:

```
$ cd server
$ bin/startup.sh
```