

How to install Documentum Core REST

Quick step-by-step guide to deploying Documentum Core REST

This summer Documentum released first version of REST services. In this guide I'm assuming you have Documentum Content Server already running in your environment. I'll try to describe everything you need in order to get your own deployment of Documentum REST.

What you need before you get going:

- 1) Documentum Content Server. Core REST 7.0 is certified with Content Server 6.7 and up, but as you will see, I'm setting it up against CS 6.6 and it is going to work just fine (at the time of writing this, I didn't have access to newer CS).
- 2) Application Server. I'm using Tomcat 6.0 but other versions of Tomcat should work equally well. If your application server runs on a different machine than Content Server, ensure machines can see each other - the easiest way would be to ping CS box from your application server machine.

That's all - pretty modest requirements as you can see.

Now download your copy of Documentum Core REST if you haven't done it already. You can find it here: https://emc.subscribenet.com/control/dctm/download?cert_num=3731241&element=4790571&dkey=NULL&plne=647381

At the above URL you'll find both: the documentation as well as the WAR file - Documentum Core REST is distributed as a **dctm-rest.war** file.

Note that it is a good practice, after you download your copy of our REST, to compare MD5 checksum with the one computed on EMC download site to ensure no corruption has taken place during transport.

Now for the actual deployment:

1. Verify that your tomcat is running before dropping our REST war file - we want to ensure that tomcat is "healthy" to begin with. For example type in the following into your browser's URL: <http://localhost:8080> - you should see Tomcat's home page displayed. Obviously adjust your URL if needed, for example your tomcat instance may be configured on a different port, etc.
2. If you got your home page in the step above, now go to next step. If not, fix your app server :-)
3. Drop **dctm-rest.war** into **webapps** folder under your root Tomcat install. On my machine this is in: E:\apache-tomcat-6.0.35\webapps
4. Since your Tomcat is running it should auto-deploy our WAR file and you should now have **dctm-rest** web app deployed and corresponding folder created under **webapps** folder. If auto-deploy didn't work on your app server, then stop Tomcat and restart it - that should do the trick.
5. Now, since we have our REST deployed, we'll need to update **dfc.properties** file: go to **dctm-rest\WEB-INF\classes** under **webapps** folder and open for editing **dfc.properties** file in your favorite text editor (I personally like Komodo Edit since it's free :-). Now set the following:
 - **dfc.docbroker.host[0]** to either host name or IP address where the docbroker is running (often it is the same machine where your Content Server is but doesn't have to be). Host name is better, in case IP address is dynamic, as long as it will resolve from your machine. It should look like the following: **dfc.docbroker.host[0]=serverA** is **serverA** was the host name.
 - ensure **docbroker.port** is correctly set, ie. **dfc.docbroker.port[0]=1489**
 - set global registry repository, ie. **dfc.globalregistry.repository=IIG_MediaLibrary** as in my environment, **IIG_MediaLibrary** is the name of my global repo
 - set global repo user name, ie. **dfc.globalregistry.username=johndoe**
 - and lastly set that user's password, **dfc.globalregistry.password=<password goes here>** (should be encrypted but for test purposes in your own environment doesn't have to be, if you don't have any sensitive data).
6. Now save **dfc.properties**, stop Tomcat and restart it, to ensure our web app can re-read new set of properties (easiest way).
7. Go back to your browser and type in full URL to our app, in my case it is: <http://localhost:8080/dctm-rest/>
8. If you followed those steps, and everything worked as it should, you should now see our Documentum REST Services Home Page. Clicking on "Services" button will execute our first REST call to bring back our first resource - link to **repositories** and **product info** resources. Default representation is in XML. If you can see that, then that confirms that your deployed REST is running fine!

That's it. We now have REST resources running. Note that our resources come both in: XML and JSON and you may want to install any necessary XML and JSON plugins to your browser to ensure that representations are formatted correctly for easier reading. From there, just follow link relations, and their corresponding HREF URLs to dive deeper into Documentum platform and see more of our REST resources. For example, follow **repositories** link relation (<http://identifiers.emc.com/linkrel/repositories>) to get list (feed) of repositories.

Please note that clicking on **repositories** link will actually fire off first remote call to the docbroker - so it is possible to misconfigure **dfc.properties** and still get REST home page correctly displayed. Following **repositories** link will actually verify your configuration.

Also note that so far you didn't have to authenticate yourself yet - only selecting given repository (by choosing one of the entries from the **repositories** feed) will prompt you for login information.

You can also try running your favorite REST client if you want something bit more sophisticated than pure browser's interface to further your interaction with Documentum Core REST.

And at this point, if you're into programming, why not checkout my sample REST mobile client for iOS, I posted as an attachment in my other blog: <https://community.emc.com/docs/DOC-26133> - all you need is Mac with Xcode.

Cheers