**Installing Remote Content File Servers: (Load balanced environments for Content Server)**

**In nut shell!!**

**Ist part 🡪 Primary Server: (People who are not aware of installing Content server please go through the installation guide Pg No 31 CS 6.5)**

1. Install the content server and configure docbase on the primary server
2. Make sure your docbase is up and running
3. Check whether you have server configs object/acs configs object are created

**2nd part 🡪Secondary server (Load Balanced Environment)**

The process is accomplished by running the CFS Configuration which is included with a Content Server installation.

1. Install Content Server on second Server
2. Configure a docbroker on the second Server.
3. Copy the server.ini file from primary to secondary server.

- Create an ODBC on the second server with the same name as was used on the primary server

- If using oracle, copy over the tnsnames.ora file from the primary.

1. Run the CFS Configuration utility.

Navigate to %DM\_HOME%\install on:

**On windows:**  Double‑click cfsConfigurationProgram.exe.

**On UNIX or Linux**, navigate to $DM\_HOME/install and type:

%dm\_launch\_cfs\_server\_config\_program.sh

When prompted for primary connection broker, specify that one which is installed on the primary server. Accept the defaults for the storage locations.

**Note: 1)** The locations can be updated later to point to a distributed storage location

5. **Once install is complete, you need to modify the server.ini i files on both servers to cross project**

**To recap, Blank or proximity = 1 means local system.2 or higher means remote. So you should always have a proximity higher then 1 for a remote connection broker**

**Server 1: Primary server**

[DOCBROKER\_PROJECTION\_TARGET]  
host = sys1  
port = 1489

 [DOCBROKER\_PROJECTION\_TARGET\_1]

host = sys2  
port = 1489  
proximity = 10

**Server 2: Secondary Server.**

[DOCBROKER\_PROJECTION\_TARGET]

host = sys2  
port = 1489

[DOCBROKER\_PROJECTION\_TARGET\_1]

host = sys1  
port = 1489  
proximity = 10

6. Once both docbases are restarted, then you should be able to see the two server configs running in DA.

7. Now you need to update the storage location to have both servers point to the same store. In this type of situation, it is recommended to create a distributed storage location to hold the data. You should also not use filestore\_01 as the default file store in this scenario just to be safe with regards to backwards and future compatibility. There is information in the Distributed guide pertaining to this as well. But basically, you already have a UNC path for your location. So you would want to do the following in DA:

a. Go to storage and create a new filestore – pointing to the UNC path you have already set for filestore\_01

b. Go to Storage and create a distributed storage location and then add the filestore from above as a component

c. Go to types and for dm\_sysobject, specify the distributed storage area for the default.

d. Finally, move the objects currently in filestore\_01 to the distributed store.

**Note:** 2) There are objects in filestore\_01 created by default during Content Server installation that are required to be in the distributed store’s component storage areas in order for EMC Documentum Desktop to function properly. However, you cannot simply make filestore\_01 a component of the distributed store to resolve this because doing so does not reset the attributes in the objects correctly and, additionally, it is strongly recommended that filestore\_01 not be a component.

Use a DQL UPDATE...OBJECTS statement to move the objects:

**DQL> UPDATE dm\_sysobject (ALL) OBJECTS SET a\_storage\_type = 'name\_of\_distributed\_store' WHERE a\_storage\_type = 'filestore\_01' name\_of\_distributed\_store is the name of the distributed storage area you created**

**Note:**

1. To login to a specific server config in DA, you can show more options on the login screen and it should then allow you to change any running server to the one you want to access. As mentioned, there is an issue with the display in DA showing the primary server as the one being logged into all the time. So if you wanted to see if the second server was actually being used, you could turn on –otrace\_authentication in the startup of each docbase. You do this in the Server Manager by clicking Edit Service. After the entry for –docbase\_name <nameofdocbase> you would simply include -otrace\_authentication. When the service is restarted, and you try logging into either server, the docbase log on that server should show the user logging in.
2. On the second server, you cannot use the Server Manager to view the docbase log or server.ini. You need to check these in the file system manually, (Documentum\dba\log and Documentum\dba\config\docbasename)
3. Make sure you have the 2 objects by firing the below DQL’s in docbase

DQL> select count(\*) from dm\_server\_config

DQL> select count(\*) from dm\_acs\_config