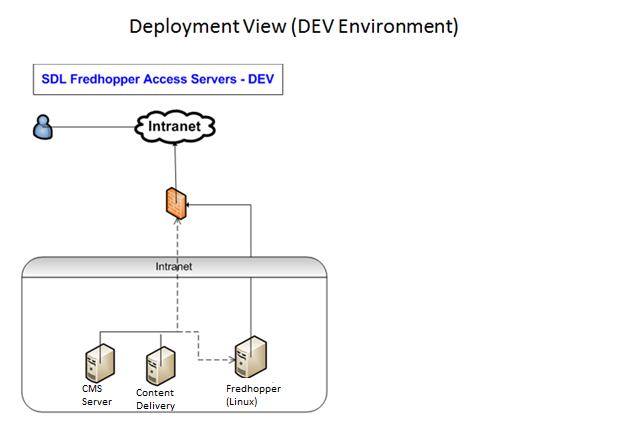
***Fredhopper/SmartTarget Installation on Linux Environment***

|  |  |
| --- | --- |
| **Submitted To:** | **Knowledge Sharing** |
| **Submitted By:** | Sumit Kumar |
| **Version:** | 1.1 |
| **Date:** | **28-April-2016** |

**INTRODUCTION** – This document is intended for the successful implementation of the SmartTarget and Fredhopper whereas Fredhopper will be implemented on Linux server and SmartTarget will be on windows server.

**Architecture:-** This is considering for the dev environment and the architecture for this depicts as below.



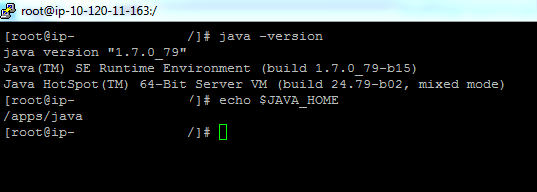
**Before starting installation**

You should have a fully-functioning SDL Tridion Content Manager and Content Delivery system

**Step 1—Installing Fredhopper**

If you are working on completely fresh Linux environment, than install JDK and set the **JAVA\_HOME** environmental variables first.

You can check the installed Java version and environment variable information.

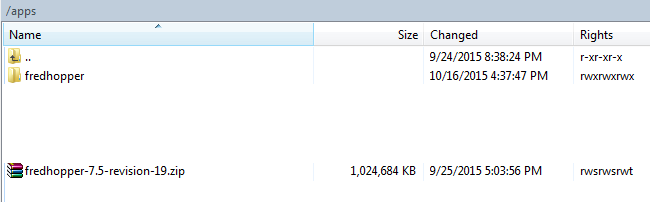


# Step 2— Install Deployment Agents

This step will install the deployment agent on all Fredhopper machines, and prepare them for use by the following steps.

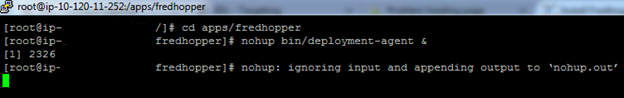
1. [Download the Fredhopper Deployment package](https://extranet.fredhopper.com/downloads/deployment/) from the Learning Center and save it
2. Log into the machine as user **fredhopper** or **superuser/root** user(as in this case)
3. cd to the **fredhopper** home directory
4. Install Fredhopper Platform:

**unzip fredhopper-platform.zip**

****

1. Start the Deployment Agent:

nohup bin/deployment-agent &

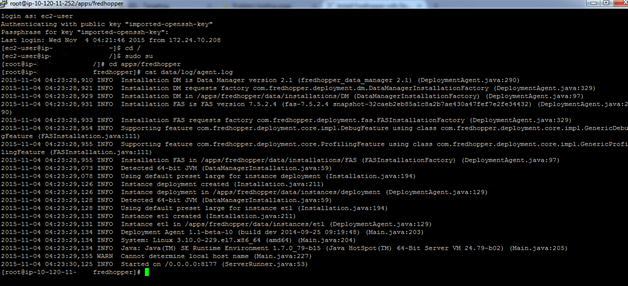
****

1. View the agent log data/log/agent.log to verify that the agent is running

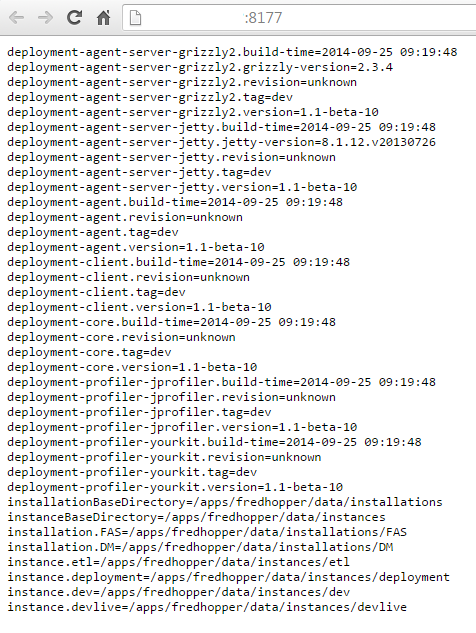
cat data/log/agent.log

If the result looks like below then everything is fine till now:-

**yyyy-mm-dd HH:MM:SS,mmm INFO  Started on /0.0.0.0:8177 (ServerRunner.java:49)**

****

1. Open in a web browser <http://localhost:8177>
2. After a successful installation, you will see in the browser properties like the installation directory and version of the deployment agent as shown in the below screenshot

****

**Note :-** The deployment-agent process must remaining running, both throughout the installation as well as when adjustments are made to your site in the future.

# Step 3: Prepare indexer machine

Execute the following steps on the indexer machine to make some adjustment in the topology file as per your naming convention:

1. Log into the indexer machine(Linux) as user *superuser*
2. Create base configuration:

cp config/topology.default.txt config/topology.txt

cp config/fasrc.default config/fasrc

1. Describe the cluster topology in *config/topology.txt*, e.g.:

**# instance name | host | preset | indexer | comment**

**dev|10.120.xx.xxx|1|-|Indexer and Preview environment for dev setup**

**devlive|10.120.xx.xxx|2|dev|Live Query Server for dev setup**

* Dev is the instance name here. It must be unique.
* **10.120.xx.xxx is** the hostname on which this instance will be running. Change the IP as per your server IP or put the hostname of the server here.
* Third column contains the instance number. Each instance on a single host must have a **unique** instance number.
* The fourth column contains either a dash ( - ) if the instance is an indexer, or the symbolic name of the indexer in case of a live instance (e.g. in this case we have put “-“ in the first line as **dev** is the indexer in our case. Then we use **dev** as in place of “-“as **devlive** is the query server and **dev** is the indexer.
* The last column are comments until the end of the line.
* Use only the pipe symbol ( | ) as the column separator.
* Do not use spaces in the first four columns
* Windows: note that the file config\topology.default.txt is using unix line endings. You can use either form.

|  |  |
| --- | --- |
| **Note:-** | **It is possible to use the same host and install the entire cluster on one machine.** |

1. (linux systems only) Adjust JAVA\_HOME and MAILTO in config/fasrc

|  |  |
| --- | --- |
|  | |
|  |

# Step 4: Install index and live machines

The following step will setup your cluster based on the contents of the config/topology.txt file. Errors in this step should be investigated and resolved before continuing.

Note :- Make sure the live machines are reachable by the indexer and running the deployment agent. If the indexer cannot reach the live machine, a "cannot connect" error will appear in the data\instances\deployment\log\kitchen.log file.

If this step fails you need to revert all the changes that may have been executed. In particular you need to remove all created instances on all nodes.

Next, execute the following steps **on the indexer machine** to prepare all instances according to the config/topology.txt file.

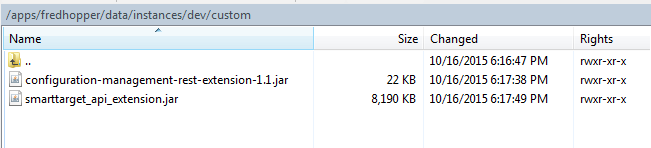
1. Change to the Fredhopper platform installation directory
2. Run the command and wait until it finishes

**bin/setup-cluster**

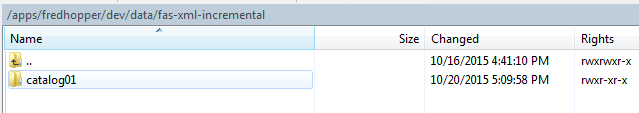
After the setup-cluster script has finished you can review the log files data/log/agent.log and data/instances/deployment/log/kitchen.log.  
These files should not contain any errors.

**grep "ERROR" data/log/agent.log data/instances/deployment/log/kitchen.log**

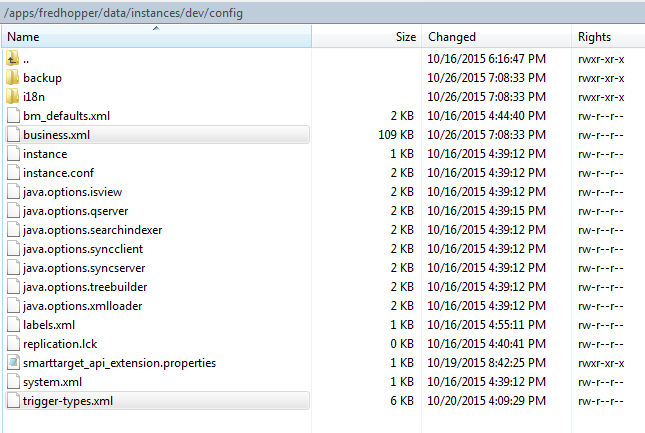
**Step 5 - Configuring Fredhopper**

1. In the /apps/fredhopper/data/instances/dev/custom folder, create a folder \custom.
2. Access the SmartTarget installation media
3. Copy configuration-management-rest-extension-1.1.jar and smarttarget\_api\_extension.jar to the \custom folder.   
   
4. For windows - In the C:\fredhopper\dev\data\fas-xml-incremental folder, create a folder \catalog01.

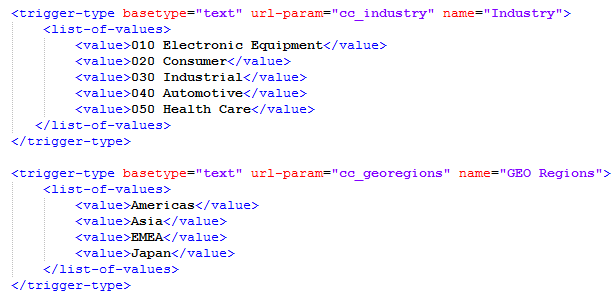
For Linux - /apps/fredhopper/demo/data/fas-xml-incremental



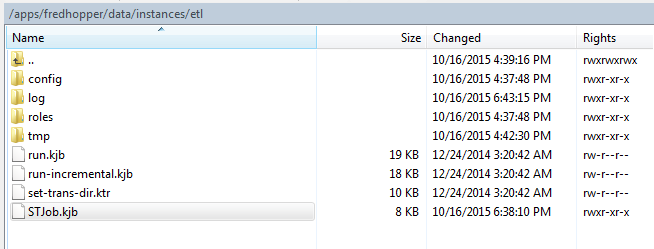
1. Copy the trigger-types.xml and business.xml to /apps/fredhopper/data/instances/dev/config and /apps/fredhopper/data/instances/devlive/config folder where **“dev”** is the indexer and **“devlive”** is the query server.



For using custom trigger you will have to add triggers as per your requirement. For example we have added the below triggers in the triggers-types.xml:-



After copied the file restart the both indexer and query server.

1. Copy the STJob.kjb file to the /apps/fredhopper/data/instances/etl folder.   
   
2. Start up the SmartTarget or SmartTargetLive instance using the command:

bin\instance SmartTarget

or

bin\instance SmartTargetLive

# Step 6: Load demo configuration

Load demo configuration (In my case I had to load this demo configuration to proceed further).

**bin/capture-import demo demo/config/demo-config-capture.zip**

In this case we have executed the below command to get the installation complete. Change the demo folder name to dev in linux directoy.

**bin/capture-import dev dev/config/demo-config-capture.zip**

# Step 7: Generate Fredhopper XML

Note:- In the below step demo should be replaced as per the indexer name, in our case it should be dev.

Create demo data in the Fredhopper XML format in the dev/data/fas-xml directory by using the "load-data" transformation. It is invoked using the run.kjb job with the "load-data" trigger identifier. This job transforms [standard CSV](https://www.fredhopper.com/learningcenter/display/learningcenter/Choose+your+Fredhopper+data+input+formats) input into Fredhopper XML. The standard CSV files are found in the directory dev/data/baseline, and when adding new ones, this is where they should be placed.

Execute the below command:-

**bin/run-etl-job run.kjb "-DINSTANCE=dev" "-DTRIGGER=load-data"**

# Step 8: Index Fredhopper XML

After the Fredhopper XML was generated it needs to be indexed.

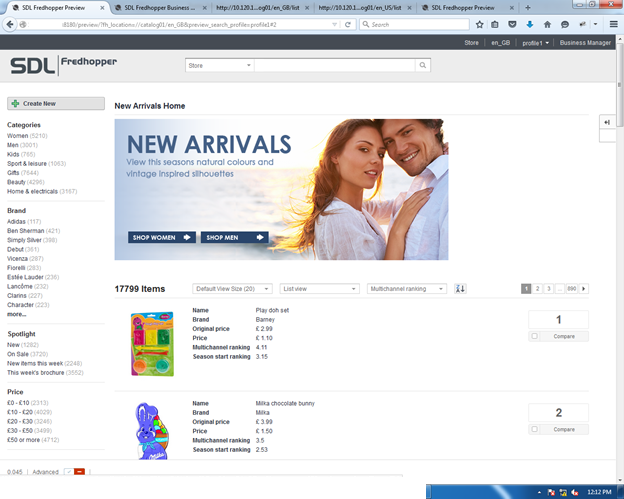
1. Start the reindex process. This will load the xml created in demo/data/fas-xml into FAS.  
   Unix:

**bin/reindex dev**

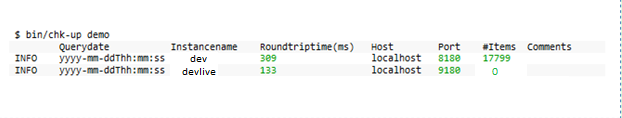
1. Open the Preview pages of the indexer instance: <http://localhost:8180/preview>, and open the Business Manager using the link at the top. The default username is admin, password admin.

**Note: The host and port will differ depending on the config/topology.txt configuration of the indexer instance.**

You should see a page like below screenshot.

****

On Unix/Linux systems an additional script is available to show the number of items loaded:

****

The last number shows the number of items the instance. Note that the bin/reindex has only loaded the items in the indexer instance.

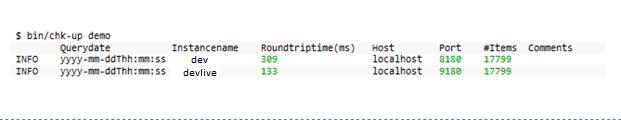
# Step 9: Publish data and configuration to live machine

Execute these steps to bring the demo to all live qservers configured in config/topology.txt:

1. File the configuration by clicking **File** in Business Manager, and then **File** again. This will open the Publication Manager.
2. Publish the configuration by clicking **Approve**, and finally, click **Publish**
3. Push the data to live (this will establish a permanent connection between the indexer and the live site, allowing for publishes in the future to be pushed to the live side):

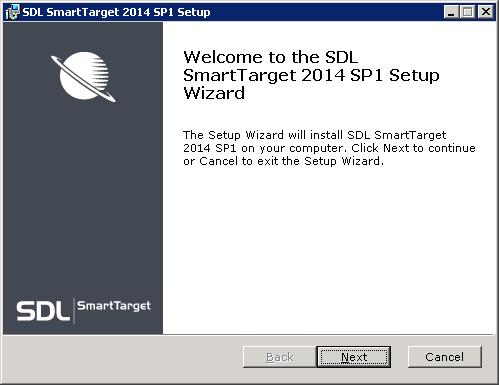
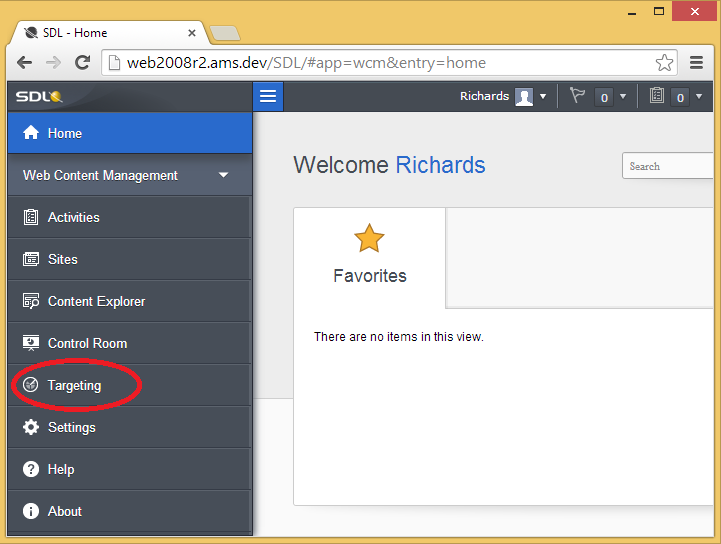
bin/fresh-index-to-live demo

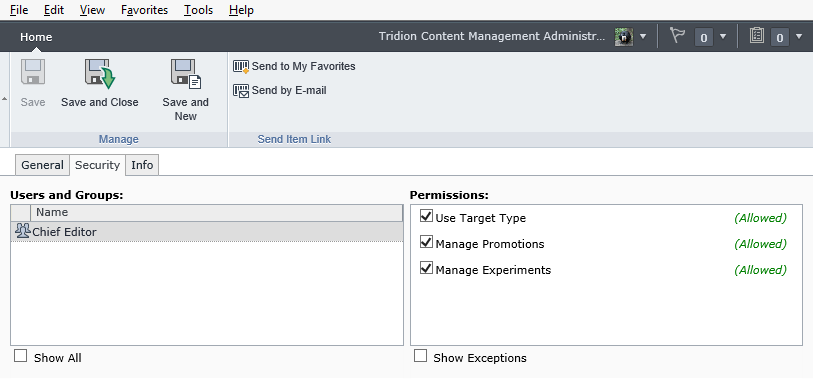
1. Open in a web browser http://<host>:<port>/preview for each live instance
2. On Unix/Linux systems you can use bin/chk-up demo now as well:

****

**Installing SmartTarget on the Content Manager**

**Procedure**

1. Access the SmartTarget installation media.
2. Browse to the Content Manager folder and double-click the Install SmartTarget 2014 SP1 shortcut.   
   
3. Follow the instructions to install (the installer does not require any inputs) and click Finish at the end. You are asked to reboot your Computer.
4. After rebooting, open your browser and you will see that the Content Manager Explorer has been updated and a new navigation item Targeting has been added to the slide out navigation.   
   
5. Target Types control the security of Publication Targets. You need to grant the below permissions : 
   * **Manage Promotions permission** :- To allow Users and User Groups to manage Promotions in the Targeting Dashboard.
   * **Manage Experiments permission**: - To allow Users and User Groups to manage Experiments in the Targeting Dashboard.



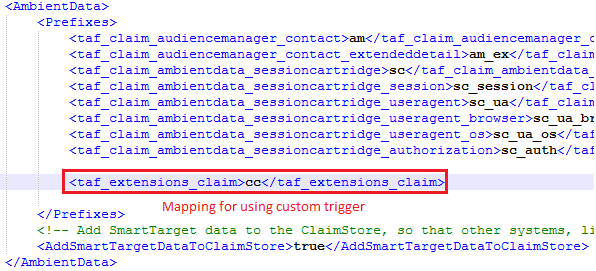
**Configuring the Content Deployer**

**Procedure**

1. Publish Components to get data into Fredhopper—the Components must be published as Dynamic Component Presentations, which you need to set in the Component Templates.
2. Open Template Builder and add the Add to SmartTarget Template Building Block (TBB) to your Component Templates to publish content to Fredhopper.
3. Open the cd\_deployer\_conf.xml
4. Add SmartTarget modules to the <Processors> element as shown below:-

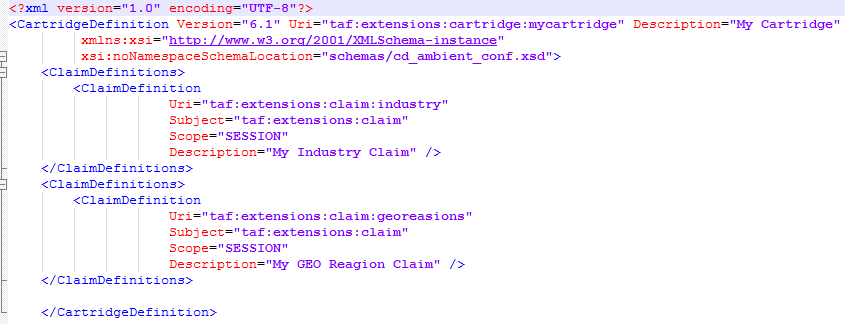


1. Save and close cd\_deployer\_conf.xml.
2. Copy the following SmartTarget JAR files to your /lib folder: 
   * smarttarget\_core.jar
   * smarttarget\_entitymodel.jar
3. Copy the following third-party JAR files to your /lib folder:
   * commons-codec.jar
   * commons-discovery.jar
   * commons-httpclient.jar
   * commons-lang.jar
   * commons-logging.jar
   * jdom.jar
   * jersey-client.jar
   * jsr181-api.jar
   * oro.jar
4. Copy the smarttarget\_config.xml to the config folder. Make sure the file is configured to point to the Indexer Server (you need to specify where the Content Deployer needs to put the XML packages for Fredhopper).   
   
5. For using custom trigger, you will be required to map the trigger variable to the smartatrget\_config.xml file as given below:-



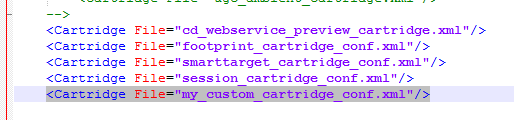
1. **Using custom cartridge: -** You will have to create custom cartridge for using SmartTarget with custom triggers. You’ll have to put this file wherever the cd\_ambient\_conf.xml file is existing.

This file contains the ClaimDefination according which the promotions will be rendered on the page.



1. **Using cd\_ambient\_conf.xml**

**You will have to give the reference for the cartridge name as shown below:-**



1. Configure your logback file.

**Installing the SmartTarget Web service**

The SmartTarget Web service is what is used for the communication between Tridion Content Manager and Fredhopper's index server.

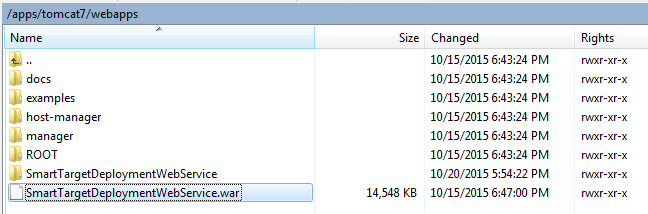
**Procedure**

1. Access the SDL Tridion installation media.
2. Depending on your preferred technology:
   * On a .NET platform, unzip the Web service package and create a new Web site in IIS
   * On a Java platform, deploy the WAR file from the roles folders.

**In this case we are using a Java Platform as Fredhopper is on different server on Linux environment and we have to deploy the SmartTarget Deployment Webservice by a .war file using Tomcat.**

**How to deploy webservice on Linux using Tomcat**

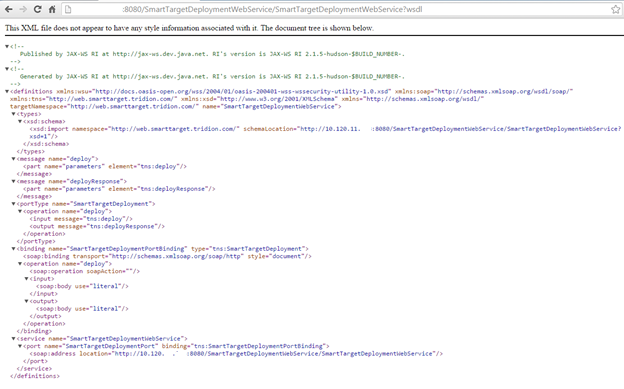
Just copy the SmartTargetDeploymentWebService.war from the installation media to the webapps directory under Tomcat and restart the Tomcat webapp server. War file will be extracted and it will create webservice application which should be accessible from Content Delivery server and Content Manager Server as well.



After successful implementation of webservice hit the URL like below.

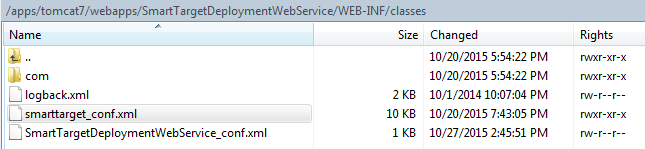
<http://ServerIP:8080/SmartTargetDeploymentWebService/SmartTargetDeploymentWebService?wsdl>

You will be see the below given screen.



**Configuration Changes:-**

Make the necessary changes to the SmartTargetDeploymentWebService\_conf.xml and logback.xml and copy the smarttarget\_conf.xml to the **“/apps/tomcat7/webapps/SmartTargetDeploymentWebService/WEB-INF/classes”**



**Setting up the Web site (.NET Web Application)**

**Procedure**

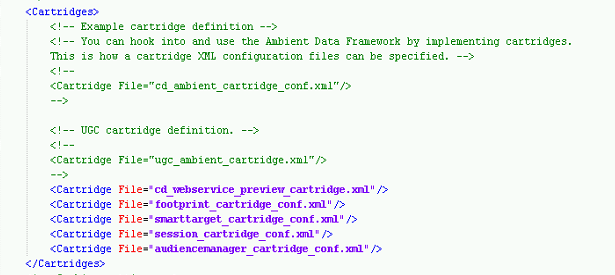
1. Go to your staging Web site.
2. Copy the following SmartTarget cartridge files to your /lib folder: 
   * session\_cartridge.jar
   * smarttarget\_cartridge.jar
3. Copy the following SmartTarget JAR files to your /lib folder:
   * smarttarget\_core.jar
   * smarttarget\_entitymodel.jar
4. Copy the following third-party JAR files to your /lib folder:
   * commons-codec.jar
   * commons-discovery.jar
   * commons-httpclient.jar
   * commons-lang.jar
   * commons-logging.jar
   * jdom.jar
   * jersey-client.jar
   * jsr181-api.jar
   * oro.jar
5. If you are using IIS, copy the following DLLs:
   * Tridion.SmartTarget.dll
   * Tridion.Smarttarget.Interop.dll

Then open the Web application web.config file and add the following line to the pages, controls element:

<add tagPrefix="SmartTarget" namespace="Tridion.SmartTarget.Web.UI" assembly="Tridion.SmartTarget" />

1. Copy across your smarttarget\_config.xml file to your config folder (your Web site needs to know where to query Fredhopper).

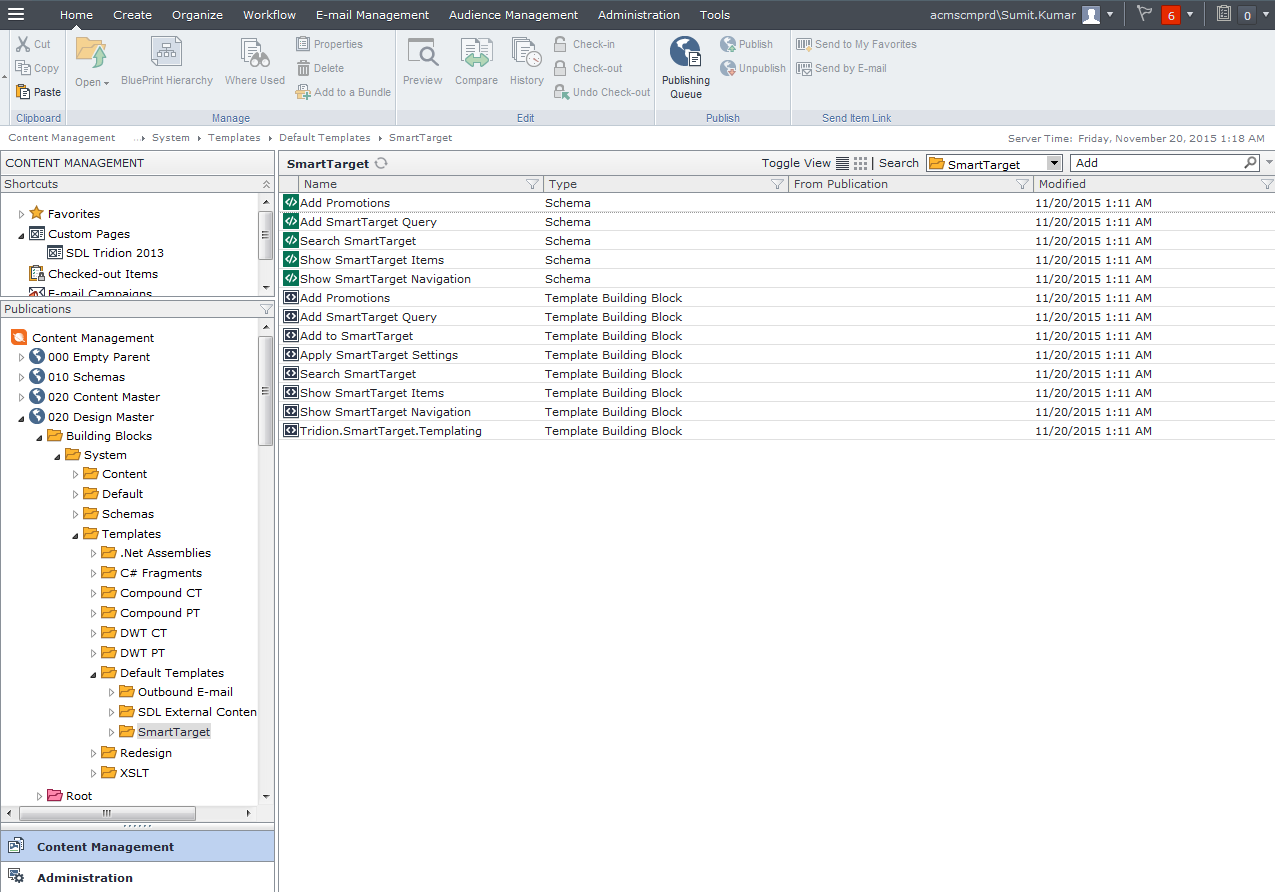
Note: In the above example, 9180 is the devlive instance as set up in the topology.txt. If you are just setting up test machine, you can specify 8180 in the smarttarget\_config.xml, which means you will actually using one instance (one instance as an Indexer and Query Server and therefore you do not have to start up devlive instances). Obviously, this depends on the type of system you want to set up.

1. Configure your logback.xml file for SmartTarget log as we did in earlier step.
2. In the cd\_ambient\_conf.xml, add references to the cartridges.   
   

**Inheriting Default SmartTarget Templates or TBBs**

When you are going to create the Promotions first time, you will need to add some SmartTarget related TBBs to you Component Templates and Page Templates. But you will have to load the SmartTarget TBBs in CMS at some base location before you start your work.

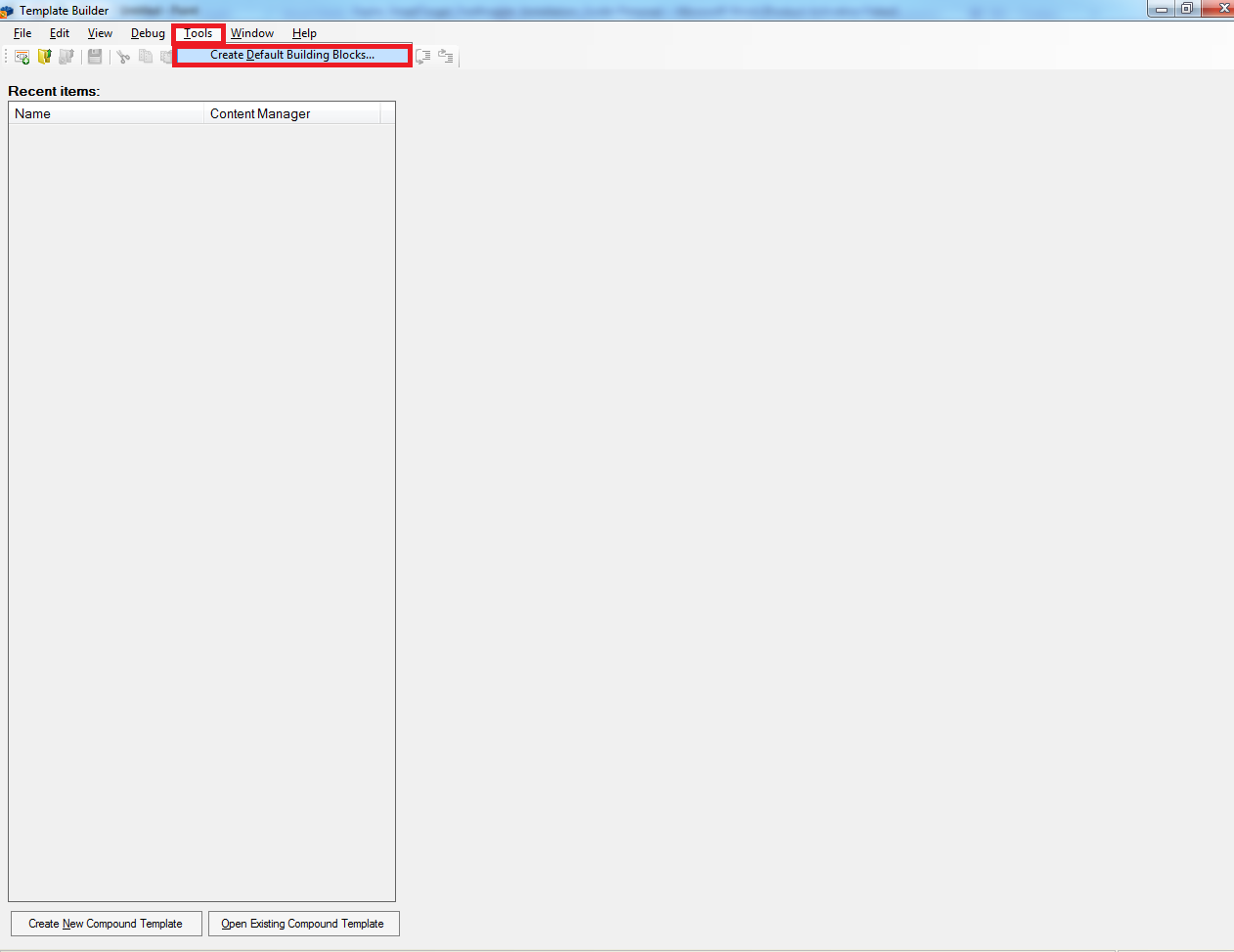
In our case we have loaded the SmartTarget Default Templates in 020 Design Master as shown below:-



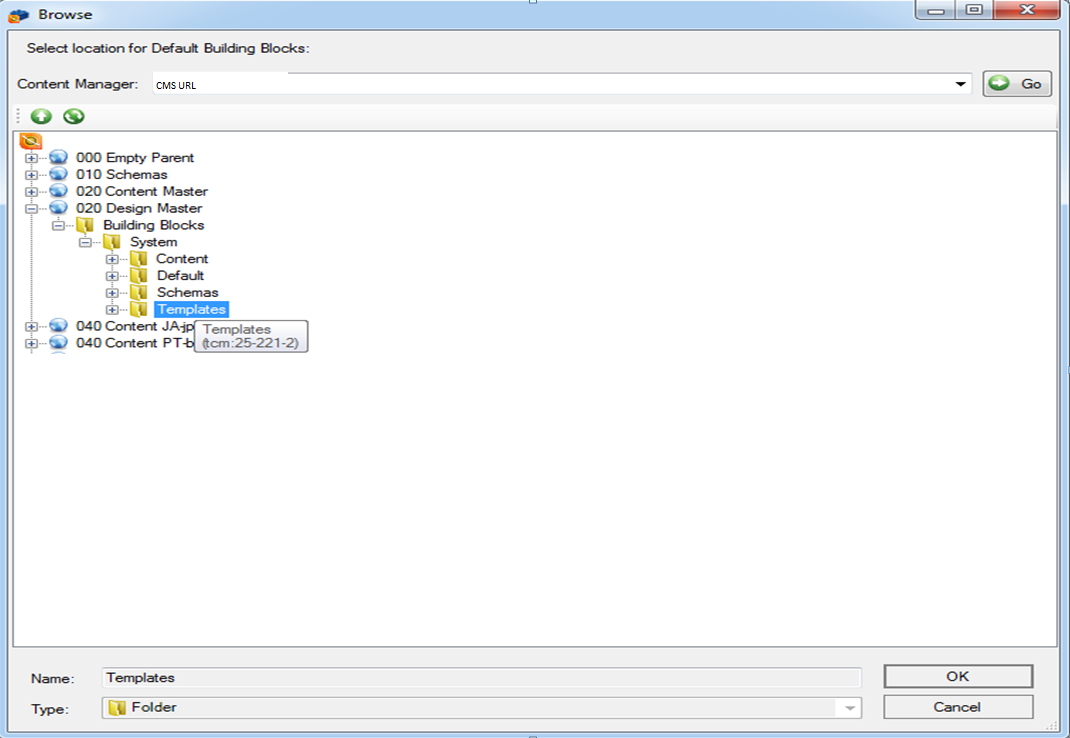
Now question is how would you do that?

Below is the process to get the Default Templates loaded in CMS.

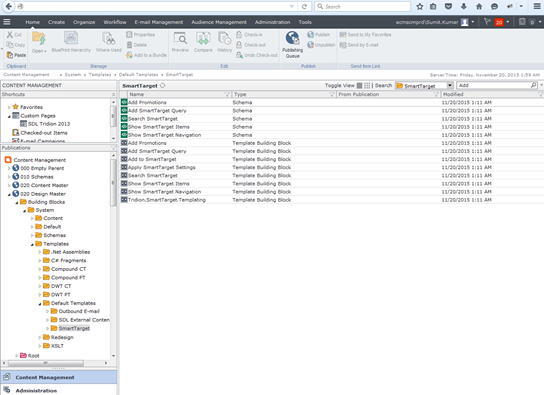
**Step 1-** Open the Template Builder, Go to Tools



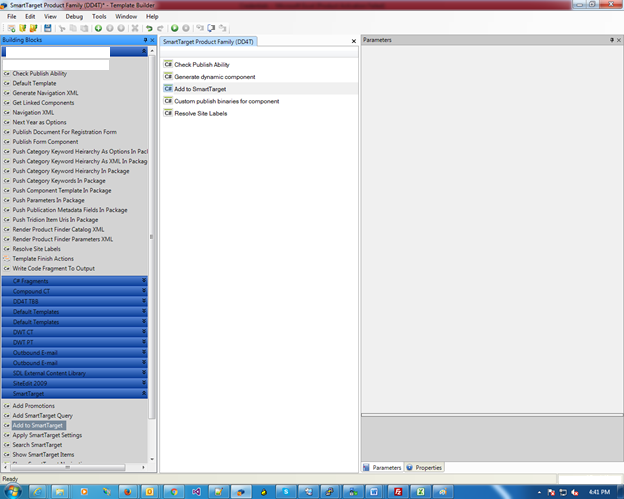
**Step 2:-** Select the folder where you want to load the Default Templates in CMS. Make sure that the folder should be accessible in all the contents.



Below is the screenshot depicting the default templates for SmartTarget.



Now you will be able to see the SmartTarget tab in the left pane (Blue part) from where you can just drag the “Add to SmartTarget” tbb on to the Component Template.



**Creating SmartTarget Promotions using DWT**

Steps for Smart Target static implementation:

1. **Create PT for page**
   1. Add Extract Components from Page tbb.
   2. Create DWT(Suppose name as ST Region DWT) having content as

<tcdl:region id="Sidebar" type="SmartTarget">

</tcdl:region>

* 1. Default Finish Actions
  2. Open Template Building Block to create PT for page(Suppose name as Smart Target PT)
  3. Drag DWT “ST Region DWT” to PT “Smart Target PT”
  4. Add SmartTarget Query tbb
  5. Add Promotions tbb
  6. Add Show SmartTarget Navigation tbb
  7. Add Search SmartTarget tbb
  8. Add Show SmartTarget Items tbb
  9. Add Apply SmartTarget Settings tbb
  10. Configure parameter values of above tbbs
  11. Run PT with page and check if it is working fine

1. **Create dynamic CT for promotion component**

Create a dynamic Component Template (Suppose SendContentToFredhopper) in Template Builder which uses the “**Add to SmartTarget**” Template Building Block to send content to Fredhopper.

* + 1. Add Default DreamWeaver Component Design DWT
  1. Add “Add to SmartTarget” tbb
  2. Add Default finish action tbb
  3. Save and close CT.
  4. Add your promotion component schema(Suppose Promotion) as a linked schema

1. **Create promotion component**

Create a component (Suppose “ST Promotion”) with schema “Promotion”.

1. **Create Promotion**

Create a promotion by going to Targeting Tab of CME. Use Trigger as current date.

1. **Create page**

Create a page using PT “Smart Target PT” having component “ST Promotion” and CT “SendContentToFredhopper”.

**Creating SmartTarget Promotions using DD4T**

* First we will cover the CMS part then we will look into the MVC project for completing the task. Below are the steps related to creating components, CTs, PTs and Promotions in CMS.

As of now we have used pre-existing Page with PT, CT (As Dynamic) and Component. Below are the details for Page used:-

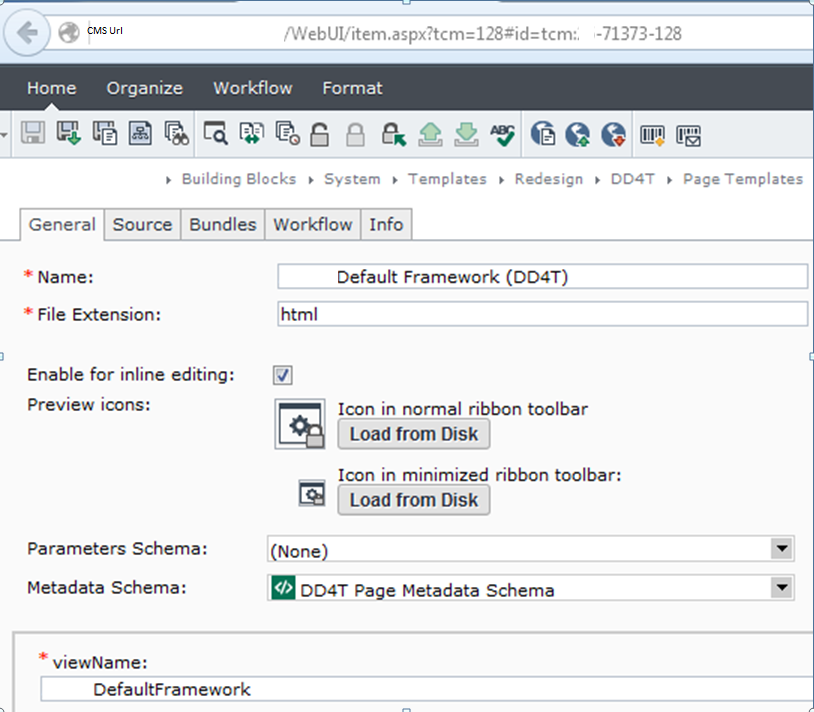
Page Name: - **STDemo(DD4T)**

File Name: - **index.html**

Rest of the details you can get referring the Page itself.

Steps for Smart Target with DD4T implementation:

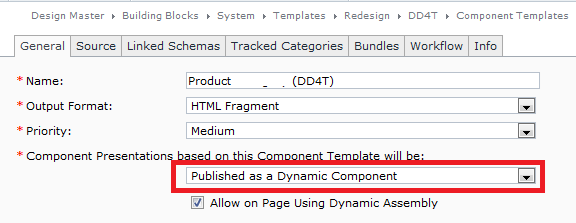
1. **Create PT for page**

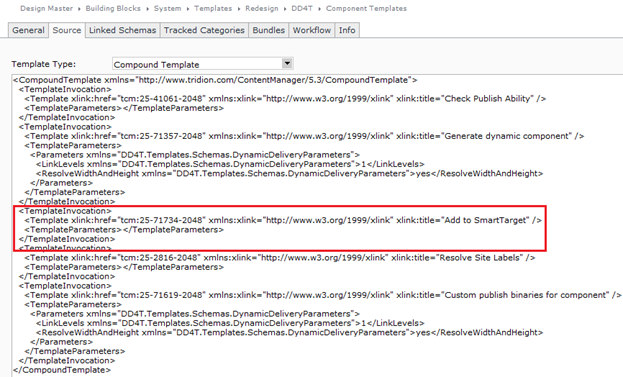


1. **Create dynamic CT for promotion component**

Create a dynamic Component Template (Product Category (DD4T) used in this case) in Template Builder which uses the “**Add to SmartTarget**” Template Building Block to send content to Fredhopper.

* 1. Add “Add to SmartTarget” tbb





1. **Create promotion component**

Here we have used existing component. Below are the details:-

For Region and Industry base promotion.

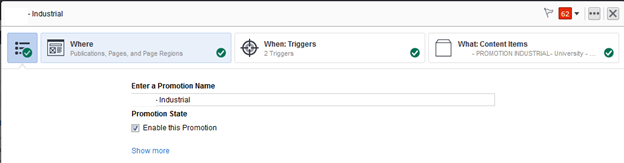
|  |  |
| --- | --- |
| Name | Demo SmartTarget Industry |
| Tcm id | tcm:36-71785 |

1. **Create Promotion**

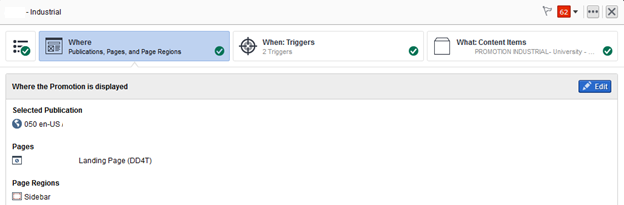
Create a promotion by going to Targeting Tab of CME. Use Trigger as Industry and Region.

Below screenshots depicts the complete process for creating the Promotion:-

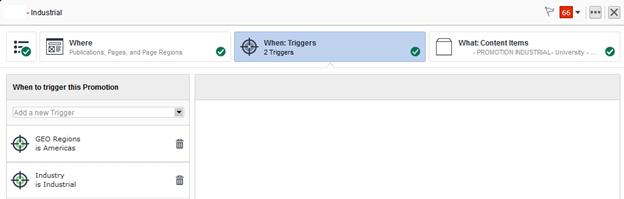
**Step 1:-**



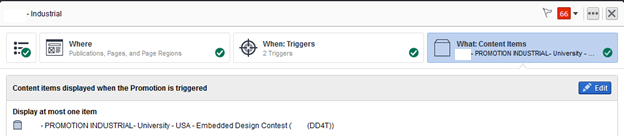
**Step 2:-**



**Step 3:-**



**Step 4:-**

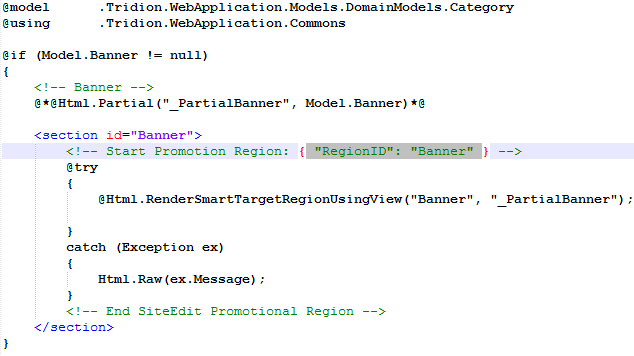


1. **Create page**

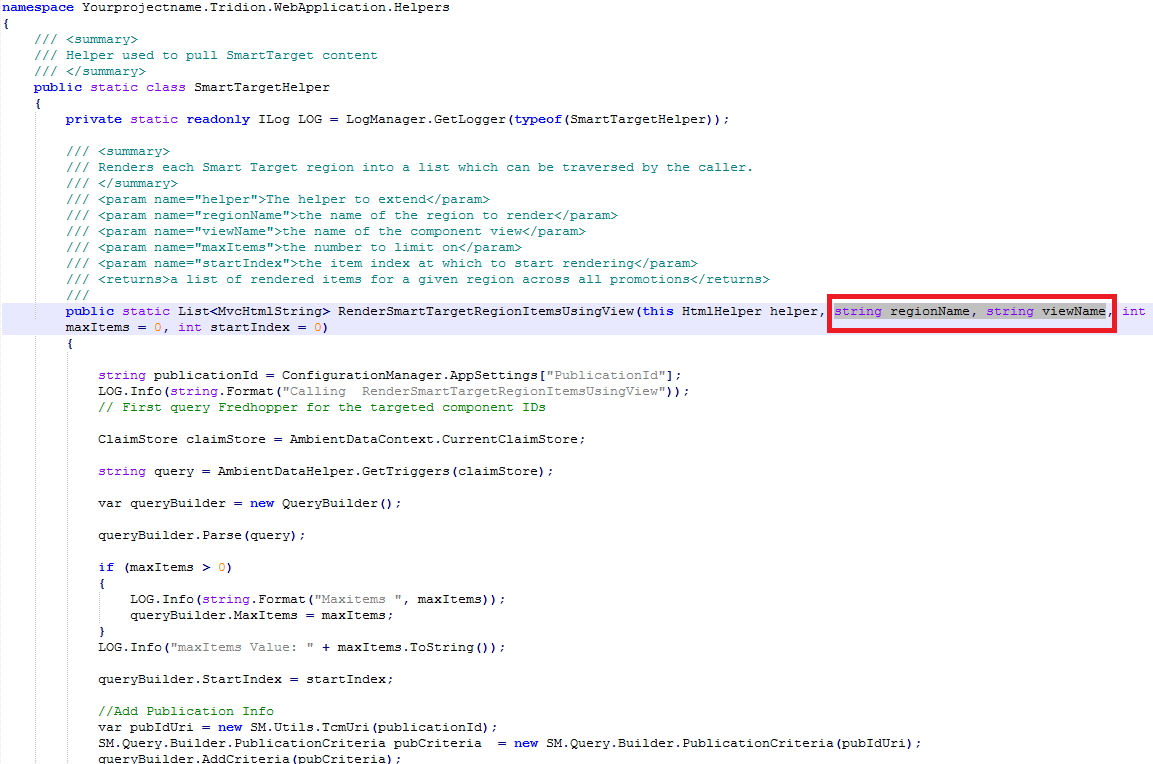
Create a page using PT (Default Framework (DD4T)) having component and dynamic CT.

* As part of DD4T application you must have an MVC project and you have Page view and component view already in place.

Below is the code snippet of the component view where we are going to render the promotion content via SmartTarget and Fredhopper. You have to provide the RegionID which you will be passing in the promotion in CMS.



**Use of Helper class: -** You will have to create a helper class for retrieving promotions from Fredhopper in case of DD4T application. I have attached the class file as well. Below is the screenshot for reference. You have to pass the region name (or region ID) and your view name on which the promotion is going to be rendered.

****

**Problems and proposed solutions**

**Problem 1**:- If you have different publishing server then probably you must get an error while publishing the content with SmartTarget tbb.

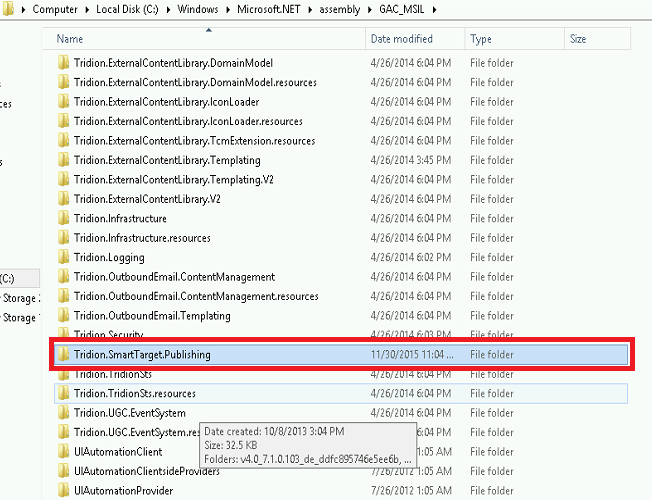
**Error**:-

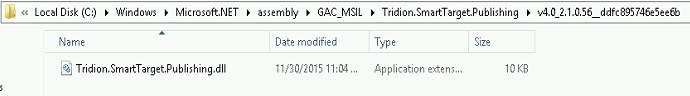
2014-03-12 11:44:32,155 ERROR DeployPipelineExecutor - Original stacktrace for transaction: tcm:0-10775-66560 com.tridion.deployer.ProcessingException: SmartTarget node not found in the componentpresentations.xml rendering metadata. Please make sure the SmartTarget publisher extension is correctly installed.

**Solution:-**

Install the SmartTarget 2014 SP1 on the publisher server as well and restart the server when the installer prompt for this.

You can check in GAC if the publisher extension is correctly installed or not. You will be able to see the below SmartTarget dll in GAC:-





**Problem 2:-** com.sun.jersey.api.client.ClientHandlerException: java.net.ConnectException: Connection refused

**Detailed error:-**

com.sun.jersey.api.client.ClientHandlerException: java.net.ConnectException: Connection refused

at com.sun.jersey.client.urlconnection.URLConnectionClientHandler.handle(URLConnectionClientHandler.java:151)

at com.sun.jersey.api.client.Client.handle(Client.java:648)

at com.sun.jersey.api.client.WebResource.handle(WebResource.java:680)

at com.sun.jersey.api.client.WebResource.access$200(WebResource.java:74)

at com.sun.jersey.api.client.WebResource$Builder.post(WebResource.java:568)

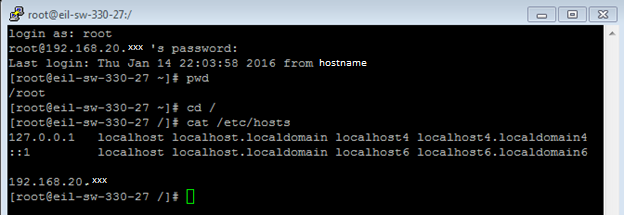
at com.fredhopper.deployment.client.impl.process.ProcessClientCommandFactory$1.execute(ProcessClientCommandFactory.java:68)

at com.fredhopper.deployment.client.CLIMain.main(CLIMain.java:309)

**Solution:-**

Check the connectivity for the IP or the hostname of the Linux machine with the port 8080 that is being used by Tomcat and port 8177 also that is being used by Deployment agent.

Check the entry of IP and hostname in the hosts file on the /ect/hosts file.



Make the entry of IP or hostname in the hosts file as shown above for 192.168.20.xxx

Note - These cases which I put here have been faced by me when working on SmartTarget with DD4T in a project. It may be possible you would not find any issue or may be different issue. So keep posting the issues and their resolution if resolved earlier to help others as well.