# Gopher Documentation

## Building Gopher :

Gopher project is structured as a maven project , where running the command **$mvn clean install** must build the required artifacts needed for the deployment. Gopher depends on two application component Floe and GoFS. So it assumes both floe and gofs artifacts are available in the local maven repository.

Follow steps below to build Gopher.

1. Build Floe. ($mvn clean install)
2. Build gofs ($mvn clean install)
3. Build Gopher ($mvn clean install)

It will create gopher-0.9.jar in the target directory.

After successfully building Floe , gofs and Gopher you should have following artifacts

**In Floe**

* flow-manager-1.0.0-SNAPSHOT-bin.zip (in modules\distribution\manager\target)
* flow-coordinator-1.0.0-SNAPSHOT-bin.zip (in modules\distribution\ coordinator \target)
* flow-container-1.0.0-SNAPSHOT-bin.zip (in modules\distribution\ container \target)

**In GoFS**

* gofs-core-0.9.jar (in modules\gofs-core\target)
* gofs-core-0.9-bin.zip (in modules\gofs-core\target)

**In Gopher**

* gopher-0.9.jar (in target)

## Deploying Gopher

Following section guides you through gopher deployment

**Deploy Floe :**

* Copy flow-manager-1.0.0-SNAPSHOT-bin.zip and flow-coordinator-1.0.0-SNAPSHOT-bin.zip to two different computers (Coping them to same computer will do not harm).
* Extract flow-manager-1.0.0-SNAPSHOT-bin.zip and go to flow-manager-1.0.0-SNAPSHOT\bin directory and run manager by running the command $ manager.bat
* Extract flow-coordinator-1.0.0-SNAPSHOT. zip and go to flow-coordinator-1.0.0-SNAPSHOT\bin directory and run coordinator by running the command $ coordinator.bat
* Copy flow-container-1.0.0-SNAPSHOT-bin.zip to the set of machines that graph partitions are stored.

Extract the flow-container-1.0.0-SNAPSHOT-bin.zip

* Go to extracted dir and edit the conf/Container.properties file changing following parameters appropriately.
  + manager\_host= host name of the machine where the manager is deployed
  + coordinator\_host = host name of the machine where the coordinator is deployed
* Create a directory in extracted directory named bsp-config
* Create two files gopher.properties and manager.properties in that directory.

**gopher.properties**

#Application class name

class=edu.usc.pgroup.goffish.gopher.sample.VertCounter

#dir where graph slices are stored

slicePath=slices

#graph id

graphId=simple\_graph

#NameNode file path

nameNodeFile=NameNode1.txt

**manager.properties**

**#Number of partitions**

numberOfProcessors=4

**Install GoFS:**

* Extract gofs-core-0.9-bin.zip
* Copy gofs-core-0.9.jar to lib directory of container
* Copy all the jars in lib dir to the lib dir of the container

**Install Gopher:**

* Copy gopher-0.9.jar to lib dir of container

**Install Application :**

* Copy application jar/s to the lib dir of the container

**Start Containers**

* Start containers by going to bin dir of containers and running $container.bat

## Gopher Sample

In this section we describe how to install and run the vertex count example we provided with the code.This sample assumes graph has four partitions and they are distributed in four different machines where containers are deployed. (See the above instructions)

**Building application**

* You can build the application using $mvn clean install command which will create vert-count-0.9-SNAPSHOT.jar
* Install application in floe(see above)

**Deploying the workflow**

* Once floe Is running you can deploy the workflow defined in graph/vert-count.xml in floe by running edu.usc.pgroup.goffish.gopher.sample.client.GraphStart in the vertex-count source we have provided. Pass command line arguments : coordinator Host name and vertex-cont.xml path .
* You will get an output like follows which gives connection details to send the workflow initializing message.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Channel Info*

*Connection Info Details*

*hostAddress =* ***68.181.17.11 (HostName to connect)***

*tcpListenerPort =* ***65446 (Data Port)***

*Control Channel info*

*Connection Info Details*

*hostAddress =* ***68.181.17.11 (HostName to connect)***

*tcpListenerPort =* ***65448***  ***(Control Port)***

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

* Then run the edu.usc.pgroup.goffish.gopher.sample.client.Client with command line arguments HostName to connect , data port and control port which will initiate the workflow.
* After the end of execution each container will have a file named **vert-count.txt** which will contain the total number of vertices in the graph.