**What is Controller Command:**

The Controller commands, which contains the business logic of the application.

The structure and significance is briefly explained here as this is very important in understanding the Order status and tracking.

A controller command is a class which interacts with Web Controller directly. Once the execution is completed, it returns the name of a view task to be completed. The web controller then checks the name and invokes the correct view command. Below is the structure and important methods of a controller command.

The Interface for the controller command:

 public interface XYZControllerCmd extends ControllerCommand {

 //This is the interface which extends the OOB (Out of the Box) Controller command to inherit the //functionalities provided by WCS.

}

The implementation class will contain the whole business logic.

 public class XYZControllerCmdImpl extends ControllerCommandImpl implements

                  XYZControllerCmd {

 Public AccessVector getResources() throws ECException {

 // If you do not know if the user has authority to take action upon that particular resource

//and you need to use that resource in your class, than only you need to override this out

//of the box method to get the targeted resource.

}

public void setRequestProperties(TypedProperty reqProperties)

            throws ECApplicationException {

 // this method can be overridden and called from the calling command when you want to set some customize values to

//the request properties.

}

public void validateParameters() throws ECException {

//this method can be overridden when we want to implement and do validation on the values //passed on request properties.

}

 public void performExecute() throws ECException {

            // This method is overridden to implement our business logic. All business logic goes in this method.

 \*\* Step 1 : call method “Do One”.

\*\* Step 2 : call method “Do Two”.

\*\* Step 3 : call method “Do Three”

            // super.performExecute can be called before implementing the customize business logic or after (depending on the requirement), from the performExecute() method only.

}

}

**Once a controller command is created**, the steps needs to be performed before the command can be used:

1. The access policy needs to be run to grant access to a specific caller.
2. Entry should be made in the CMDREG table and registry should be refreshed from the Admin console OR the implementation class should be mentioned in the Interface itself using the following code :

 public static final String defaultCommandClassName=

                  “Path to the ‘XYZControllerCmdImpl ‘ “